EXHIBIT 142

REDACTED

INTERNAL ONLY

Header Bidding Observatory #3

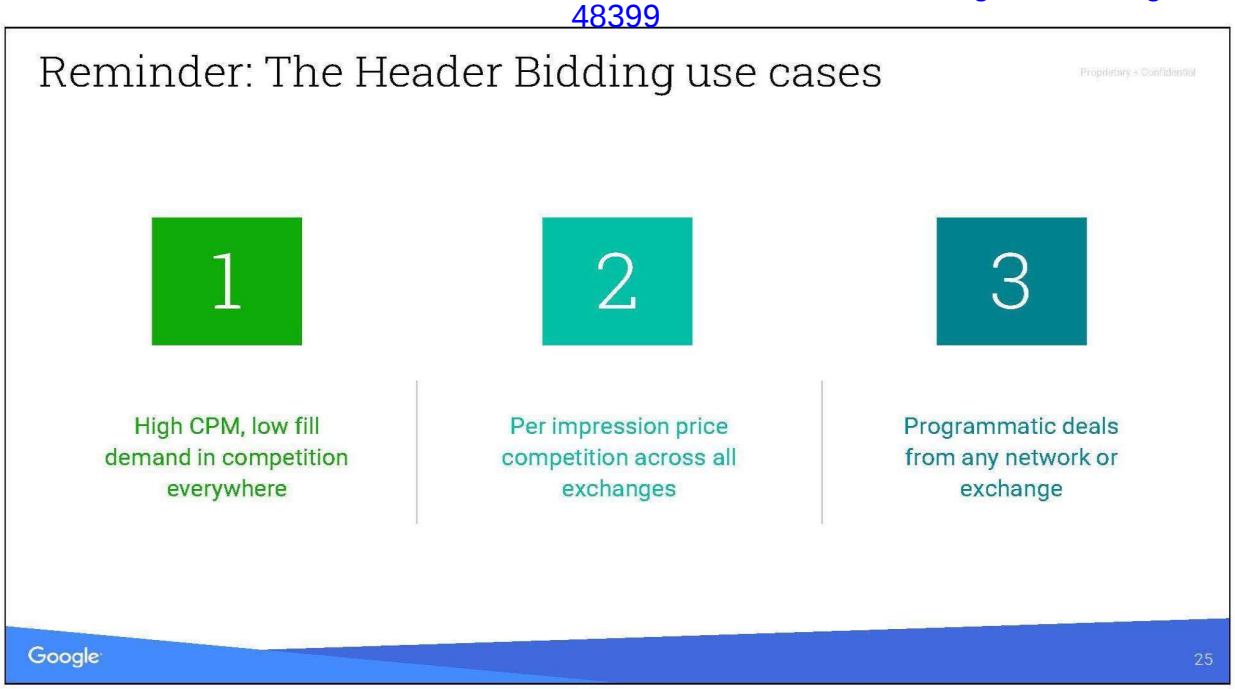
Q1 2018

[redacted]

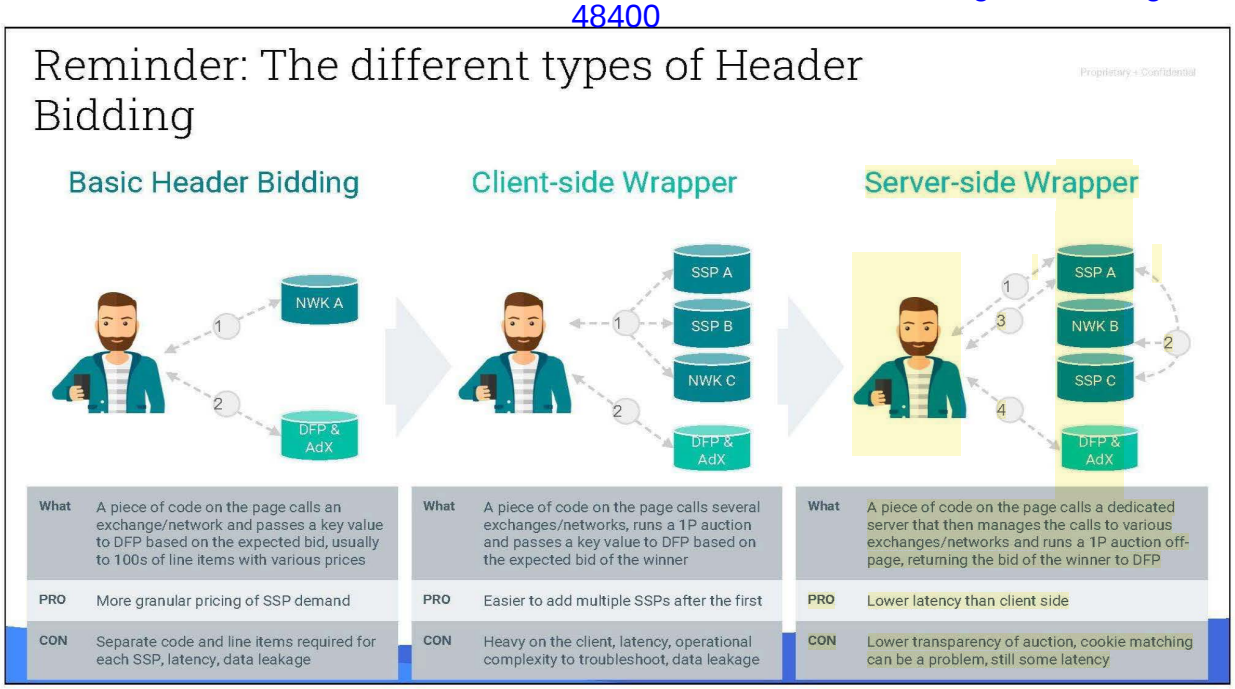
hbobservatory@

Google

Internal Only
Proprietary + Confidential



Here are the three distinct use cases have developed due to the investment in header bidding.
High CPM, low fill demand (e.g. remarketing by Criteo and Amazon) is able to compete against the vast majority of standard direct and indirect impressions
Exchanges can compete against one another with accurate per impression pricing
Programmatic deals from any network or exchange can compete against traditional directly sold ads



There are also different types of Header Bidding

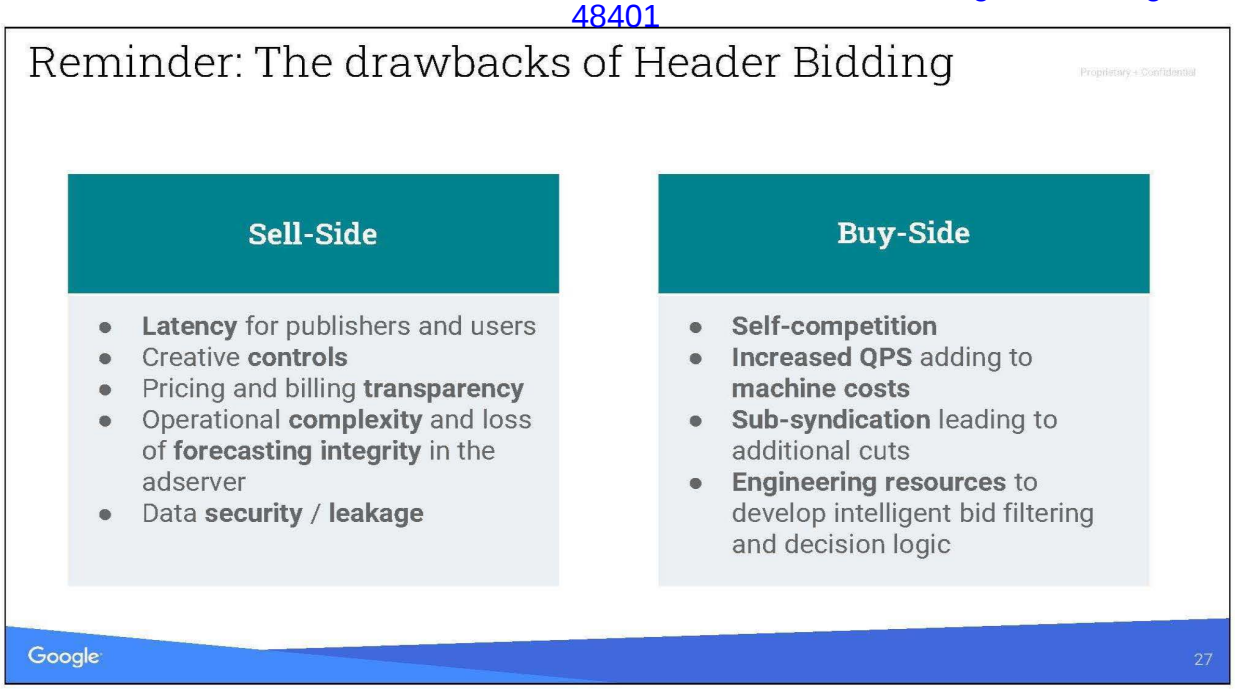
Basic Header Bidding
Simple to work with one partner but separate codes if working with several SSPa, adding latency

Client-side Wrapper
Offers inventory to multiple exchanges before making ad calls to their ad servers
Ad calls are made from the user's browser and auction rules are also run by the browser using code in the header
Prone to higher levels of latency and impacted user experience
Examples: Rubicon Fast Lane, AppNexus

Server-Side Wrapper
Ad calls are made outside of the user's browser with the auction taking place in an external server after a single ad call is made
However, the browser still makes the first call to the header bidder server and receives an answer which it passes to the adserver
Improves speed and latency, but will not without any drawbacks (e.g. potential cookie loss)
Example: Amazon, Prebid.js Server

These are distinct from true server-to-server integration like Google's Exchange Bidding, which is NOT a form of header bidding (there is no code in the header)

Server to Server
All ad calls and auction rules are run off-browser with the tech provider managing and hosting the entire solution
There is no header code on the page and no initial call from the user's browser to the header bidding server. Communications flow directly between the publisher ad server and the other server.
Allows for a true unified auction to take place, with improved speed and reduced latency and inefficiencies for partners
Example: DoubleClick Exchange Bidding in Dynamic Allocation



While the technology in Header Bidding has become more sophisticated over time, it still requires at least one extra tag on the page. The additional client-side request(s) along with other server-side technology add latency for both publishers and users. Other problems exist throughout the end-to-end process (e.g. creative controls, pricing & billing transparency, data leakage) but publishers have been willing to put up with such issues in return for greater revenue.

On the buy-side, header bidding poses a different set of issues. When multiple exchanges are called for a single ad slot, buyers wind up bidding multiple times for the same impression, resulting in self-competition. The increase in queries-per-second (QPS) for both networks and DSPs add to machine costs. These problems gets worse when header bidding providers sub-syndicated impressions to other unauthorized SSPs that take a cut and reduce the share of an advertiser's dollar that makes it to the publisher. Beyond the pure machine costs and additional revenue shares or tech fees, engineering resources are needed to develop intelligent bid filtering and decision logic.